

Un système expert pour DrGeo.

Le package Myexpert. pck.st ajoute un système expert qui fait des déductions logiques à partir d'une figure DrGeo et de règles.

C'est une adaptation d'une version Pharo, qui est en cours de développement pour CUIS smalltalk.

Pour le moment, il couvre les notions de points, segments, droites, cercles et angles.

A faire : les triangles, les quadrilatères et les polygones réguliers.

Le système expert peut aussi être utilisé pour des domaines autres que la géométrie, quelques exemples sont donnés, c'est une source de projets pour le cours d'informatique, ceci permet aussi de distinguer l'utilisation de l'IA basée sur la logique par opposition à l'IA générative basée sur les probabilités.

Du point de vue pédagogique, Erudite, le système de traitement de texte contenu dans CUIS smalltalk permet d'ajouter au *learning by doing* le *learning by example* que l'on trouve dans DrGeo avec l'éditeur de scripts et la documentation avec Erudite destiné à enseigner la programmation smalltalk.

La partie de Erudite qui concerne la programmation par script de DrGeo est complété par la documentation du système expert et de la géométrie.

La documentation du système expert décrit le système expert avec un tutoriel.

La documentation de la géométrie reprend les notions de l'Aide Mémoire de mathématique utilisé dans les cantons suisses romands.

Le système d'organisation des sections de Erudite en arbre, semblable à l'organisation des fichiers d'un ordinateur permet de retrouver facilement la section qui nous intéresse.

Ces sections de la documentation se veulent une illustration de ce que l'on peut obtenir du système expert à propos du sujet traité. C'est un point de départ pour l'exploration.

En tête de section, on trouve des définitions qui font référence aux livres de Euclide et à l'Aide Mémoire de mathématique, pour celui-ci les pages concernées sont signalées.

Le début d'une section commence en général par deux liens actifs.

L'un permet d'avoir le tracé de la figure et une forme résumée des déductions du système expert, il est destiné à une **utilisation à partir du menu de la figure**.

L'autre donne le script de la figure et une forme résumée des déductions, il est destiné à une **utilisation avec le langage smalltalk**.

Le dialogue avec le système expert se déroule dans une fenêtre espace de travail qui a l'avantage quand on utilise du code de s'occuper de la déclaration de variables.

Quand on utilise le menu, c'est avec des sélections dans l'espace de travail qu'on communique avec le système expert.

Description du système expert

Un système expert est basé sur la logique des propositions.

Une proposition qui n'a pas de variables est dite d'ordre 0.

Une proposition qui a des variables est dite d'ordre 1.

On peut faire une comparaison avec l'arithmétique qui n'a pas de variables et l'algèbre avec les équations.

On peut considérer le système expert comme un logiciel qui résout des problèmes logiques.

Les données d'un problème sont des affirmations rangées dans une base de données.

A partir de ces affirmations le système expert va ajouter de nouvelles propositions obtenue en appliquant des règles, ces nouvelles affirmations sont ajoutées à la base de données.

Structure de liste.

Les faits peuvent être une simple suite de mots mais dans ce système expert ce sont des structures de listes qui peuvent se combiner comme les expressions mathématiques avec des parenthèses, pour les listes, ce sont des crochets [].

Les règles.

Une règle applique le principe du syllogisme utilisé par les Grecs.

Les prémisses du syllogisme sont les conditions de la règle et si elles sont satisfaites, la conclusion est ajoutée à la base de données.

Si le système utilise la logique d'ordre 1, les conditions peuvent avoir des variables, en ce cas le système expert cherche en plus les valeurs des variables.

Il n'y a qu'une seule conclusion par règle et les conclusions ne figurent qu'une fois dans la base de donnée.

Les règles sont des méthodes smalltalk de la forme `lisRegle:nom si: condition alors: conclusion`.

Les variables commencent par le caractère ?.

Exemple :

```
e lisRegle: 'parents'
si: 'père ?e ?p
mère ?e ?m'
alors: 'parents ?e ?p ?m' .
com: 'Les parents d'un enfant sont son père et sa mère'.
```

Un commentaire peut être associé à la règle pour indiquer le but de la règle.

Les règles sont le savoir du système expert.

Le système expert essaie les règles et ajoute leurs conclusions à la base de données.

Si une conclusion est trouvée, le système examine à nouveau les règles.

Les modèles de question.

Les conclusions peuvent être nombreuses. C'est là qu'intervient la structure de liste.

Les modèles de questions sont les conclusions des règles résumées sous forme de listes de premier niveau.

Cela veut dire que les variables du modèle contiennent soit des mots soit des listes.

Un modèle de question est vérifié soit par la procédure **vérifieCondition**: *condition* en cas d'utilisation du code smalltalk, soit par l'article **vérifier** du menu expert après avoir sélectionné un modèle de question dans l'espace de travail.

Cette méthode ou l'article **vérifier** peuvent aussi vérifier la partie **condition** d'une règle, dans ce cas il y a une condition par ligne.

Vérifier un modèle de question révèle la structure de liste des conclusions.

On peut ensuite poser des questions plus précises au système expert en remplaçant des mots par des variables.

Exemple extrait du tutoriel.

```
"Initialiser les faits"
exp initFaits. 'Base vide' .
```

```
" Affirmer les faits."
exp affirmeFaits: 'père Pierre Jean
mère Pierre Léa'. '2 faits affirmés.' .
" Afficher les faits"
exp afficheFaits. '
1 père Pierre Jean ->énoncé
2 mère Pierre Léa ->énoncé
' .
"Initialiser les règles"
exp initRegles. 'règles supprimée' .
"Définition de la règle"
exp lisRegle: 'parents'
si:'père ?e ?p
mère ?e ?m'
alors:'parents ?e ?p ?m'
com:'Les parents d'un enfant sont son père et sa mère'. 'Règle "parents" définie' .
```

```
"Examiner les règles"
exp deduis. '
Dédutions effectuées ' .
exp modèlesQuestions. . '
Modèles de questions.
mère ?x1 ?x2
parents ?x1 ?x2 ?x3
père ?x1 ?x2
' .
```

```
"Vérification d'une condition"
exp vérifieCondition:' parents ?x1 ?x2 ?x3'. '
3 parents Pierre Jean Léa ->parents
* Les parents d'un enfant sont son père et sa mère
' .
"Une valeur à la place d'une variable"
"On cherche les parents de Pierre"
exp vérifieCondition:' parents Pierre ?x2 ?x3'. '
3 parents Pierre Jean Léa ->parents
* Les parents d'un enfant sont son père et sa mère
' .
"On cherche un enfant de Jean"
exp vérifieCondition:'père ?x1 Jean'. '
1 père Pierre Jean ->énoncé
```

Système expert et géométrie.

En exemple une des sections.

28 Squeak! (C:\Users\OmniBook Ultra\Desktop\Temp transfert\DrGeoCuisVierge\DrGeo\Resources\image\drgeo7.image)

Modifier un script Smalltalk

GéométrieSystèmeSgmnt

exp lisFigure.
Faits de la figure chargés.

exp deduis
Deductions effectuées

exp modèlesQuestions.
Modèles de questions.
DrGLine2ptsitem 7x1 7x2 7x3
DrGPointFreeitem 7x1
DrGPointOncurveitem 7x1 7x2
adjacent 7x1 7x2 7x3 7x4 7x5
alignement 7x1
couple 7x1
coupleSur 7x1 7x2
droite 7x1
droite 7x1 7x2
droite 7x1 7x2 7x3
entre 7x1 7x2 7x3 7x4
lignePolygonale 7x1
origine 7x1 7x2
parent1 7x1 7x2
parent2 7x1 7x2
parents 7x1 7x2
point 7x1
point 7x1 7x2
point 7x1 7x2 7x3 7x4 7x5
pointsSur 7x1 7x2
pointsSur 7x1 7x2
position 7x1 7x2 7x3 7x4
sens 7x1 7x2
sorteDeDroite 7x1
sorteDeDroite 7x1 7x2 7x3
triangle 7x1 7x2

vérifications: droite 7x1 7x2 7x3
11 droite dAB A B ->interface-Droite
* Lecture d'un fait sur la figure

vérifications: droite 7x1 7x2 7x3
pointsSur:rep: 7x1 7pts
11 droite dAB A B ->interface-Droite
* Lecture d'un fait sur la figure
pointsSur:rep: dAB [A P B]
"la liste de points sur un droite ordonnés
suivant le sens de la droite"

Introduction
Exemples
Point
Général
Vecteur
Lignes droites
Lignes courbes
Numérique & Texte
Transformations
Styles
Expert
Introduction
Utilisation interactive
Tutoriel
Liste des prédicats
Exemples d'experts
Géométrie et système Expert
1 point segment et droite
1.1 Point
1.2 Segment
1.3 Droite
1.3.1 Demi-droite
1.3.2 Intersection de dro
2 Cercles.

1.2.2. Droites
1.2. Segments et droites.
Euclide : Définition 2, « **Une ligne est une longueur sans largeur.** »
Euclide : Définition 3, « **Les extrémités d'une ligne sont des points.** »
Euclide : Définition 4, « **La ligne droite est celle qui est également placée entre ses points.** »
Si on marche le long d'une droite le pied gauche se déplace de la même distance que le pied droit.
Sur une sphère les méridiens sont des droites mais pas les parallèles, les pas ne sont pas de même taille de chaque côté d'un parallèle.Tous les parallèles sont de taille différente.)
Euclide : Demande 1, « **Entre deux points on peut toujours tracer une droite**»(Euclide défini comme droite ce qui est pour nous un segment.)
Euclide : Demande 2, « **On peut toujours prolonger indéfiniment une droite tracée entre deux points.**»
(Euclide demande ce qui est pour nous une droite. Cette demande est toujours satisfaite pour la géométrie plane, elle n'est pas toujours vérifiée, par exemple, sur un sphère, les extrémités d'un segment peuvent se rejoindre en un même point.)
Aide-mémoire p.90 : « **La droite A B est constituée d'une infinité de points alignés avec A et B** »
Aide-mémoire p.90 : « **Un segment est une portion de droite limitée par deux points.** »
Script de la figure
[Charger la figure et les modèles de questions](#)
[copier le code de la figure](#)
Le prédicat pointsSur:rep: permet d'avoir la liste des points sur une droite.
Les points sont ordonnés dans le sens de la droite.
droite 7x1 7x2 7x3
pointsSur:rep: 7x1 7pts
[copier pour vérifier avec le menu Expert](#)
exp verifieCondition: 'droite 7x1 7x2 7x3
pointsSur:rep: 7x1 7pts'.
[copier l'exemple](#)

Sélectio
inner.at